New Authentication Plugin

PARSEC

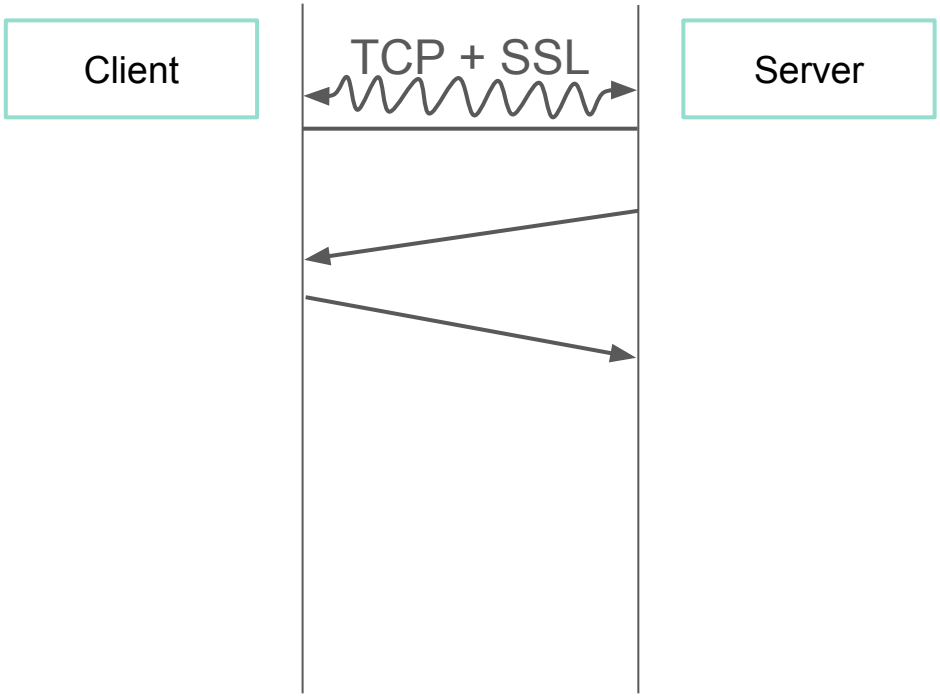
Nikita Maliavin

Zero-configuration

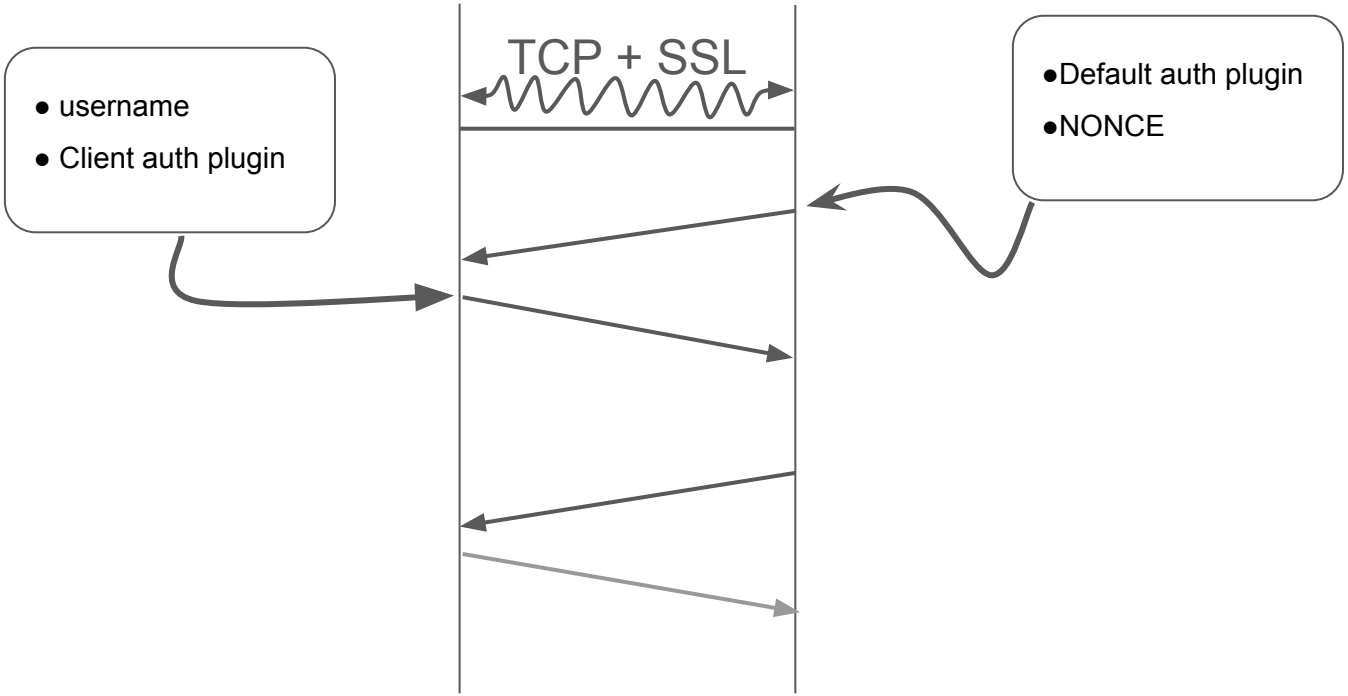
SSL

Sergei Golubchik

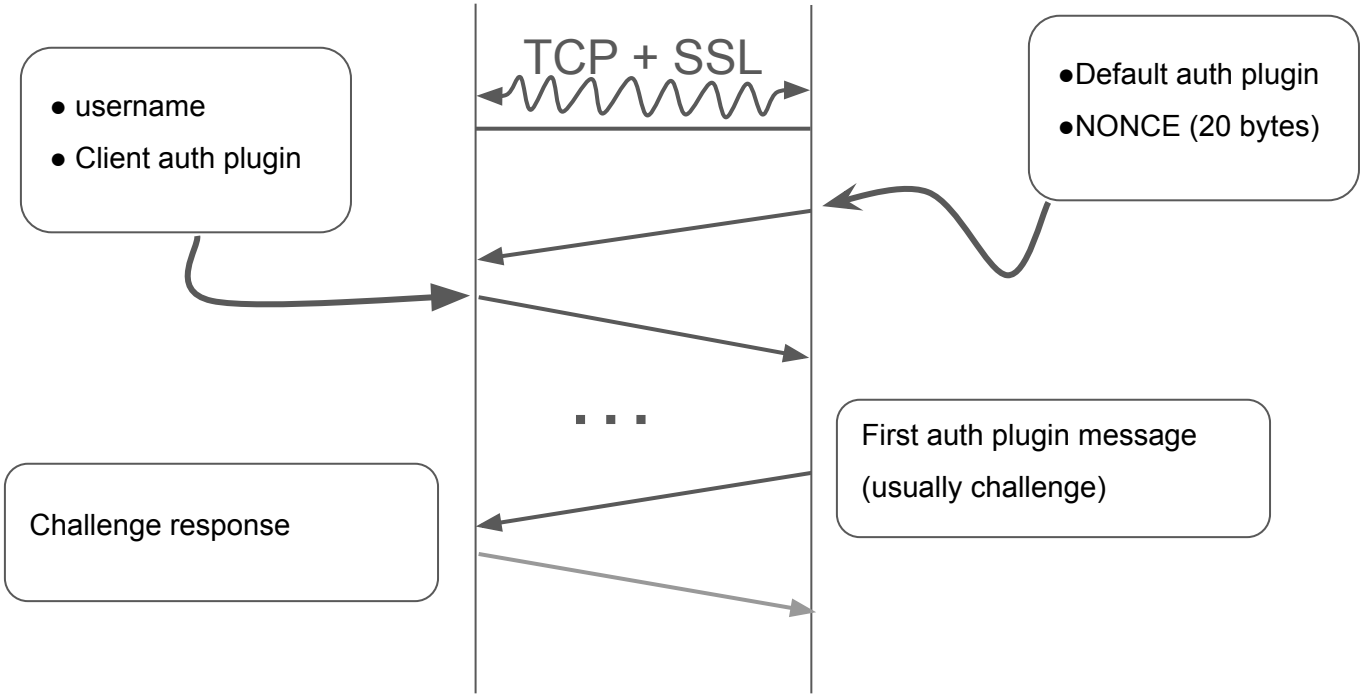
Authentication plugins?



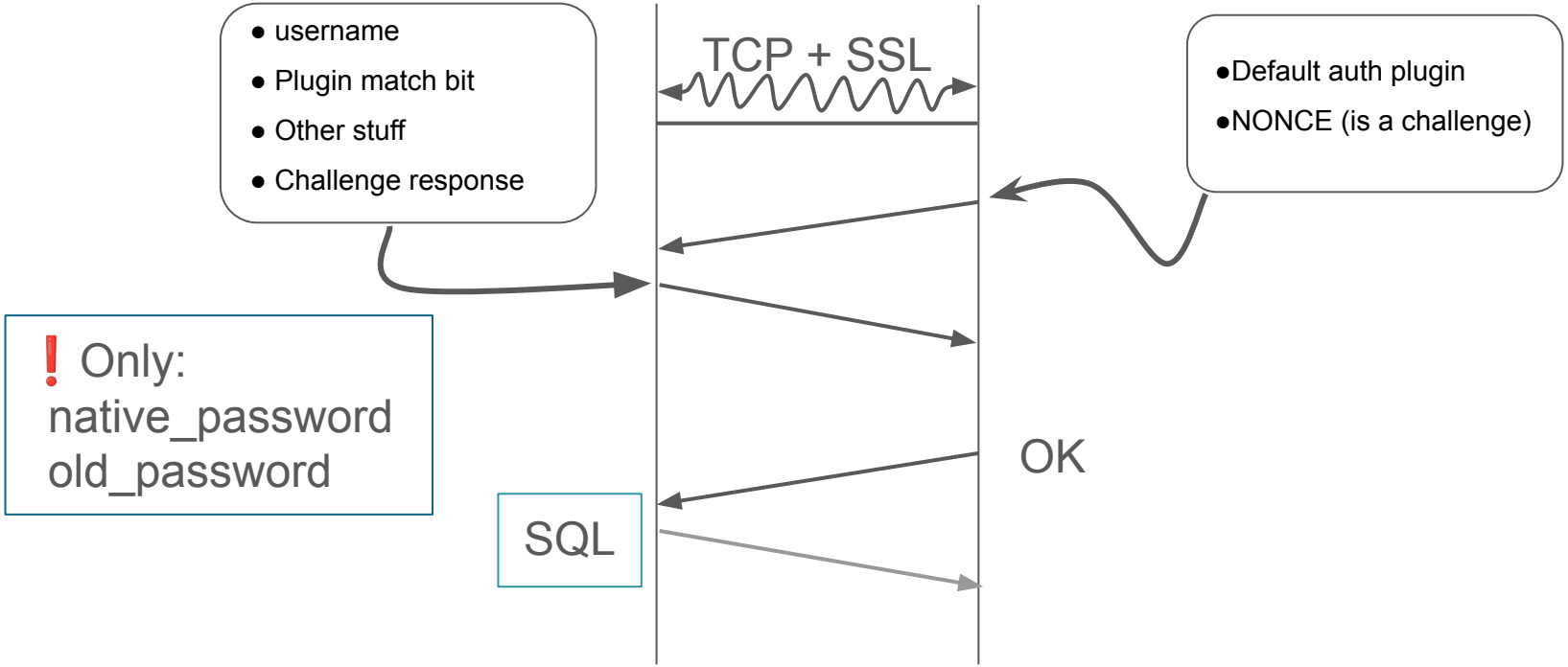
Authentication plugins?



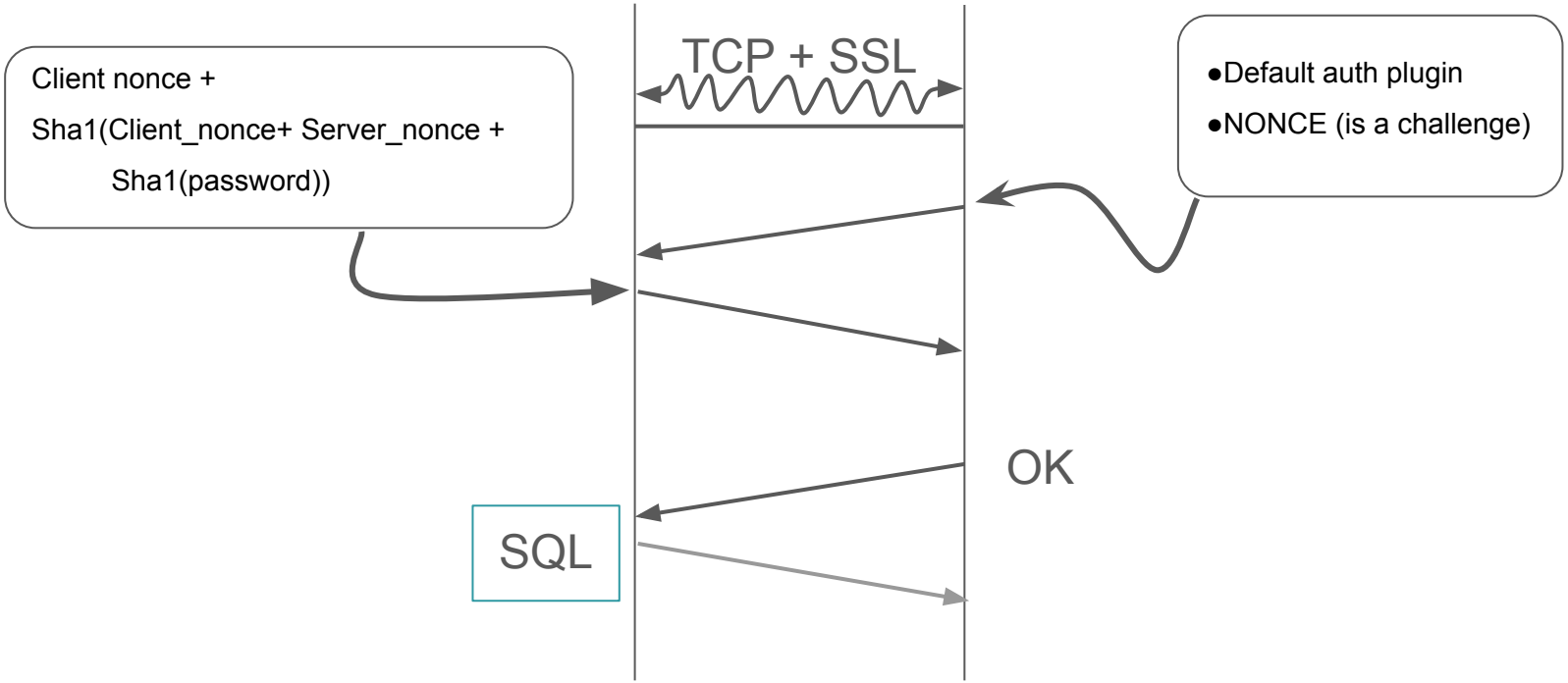
Authentication plugins!



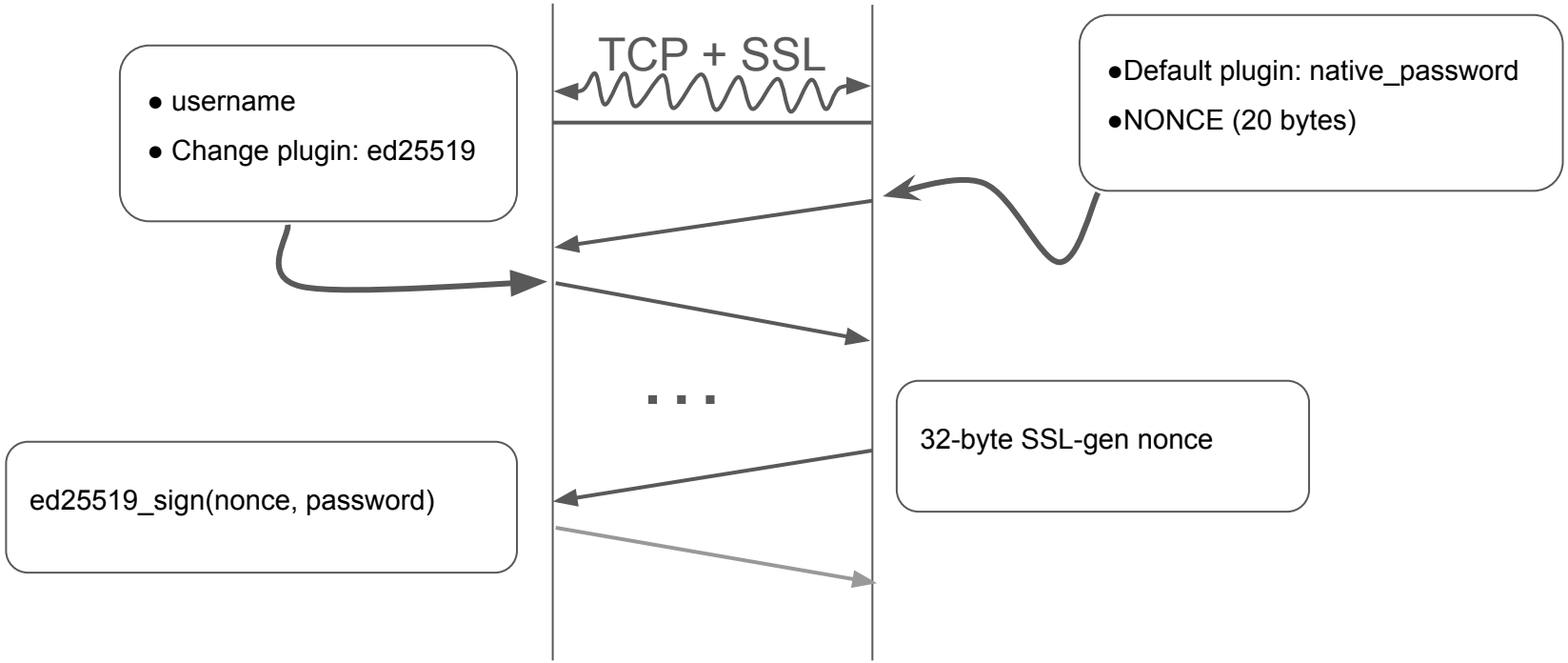
Good case: first try match



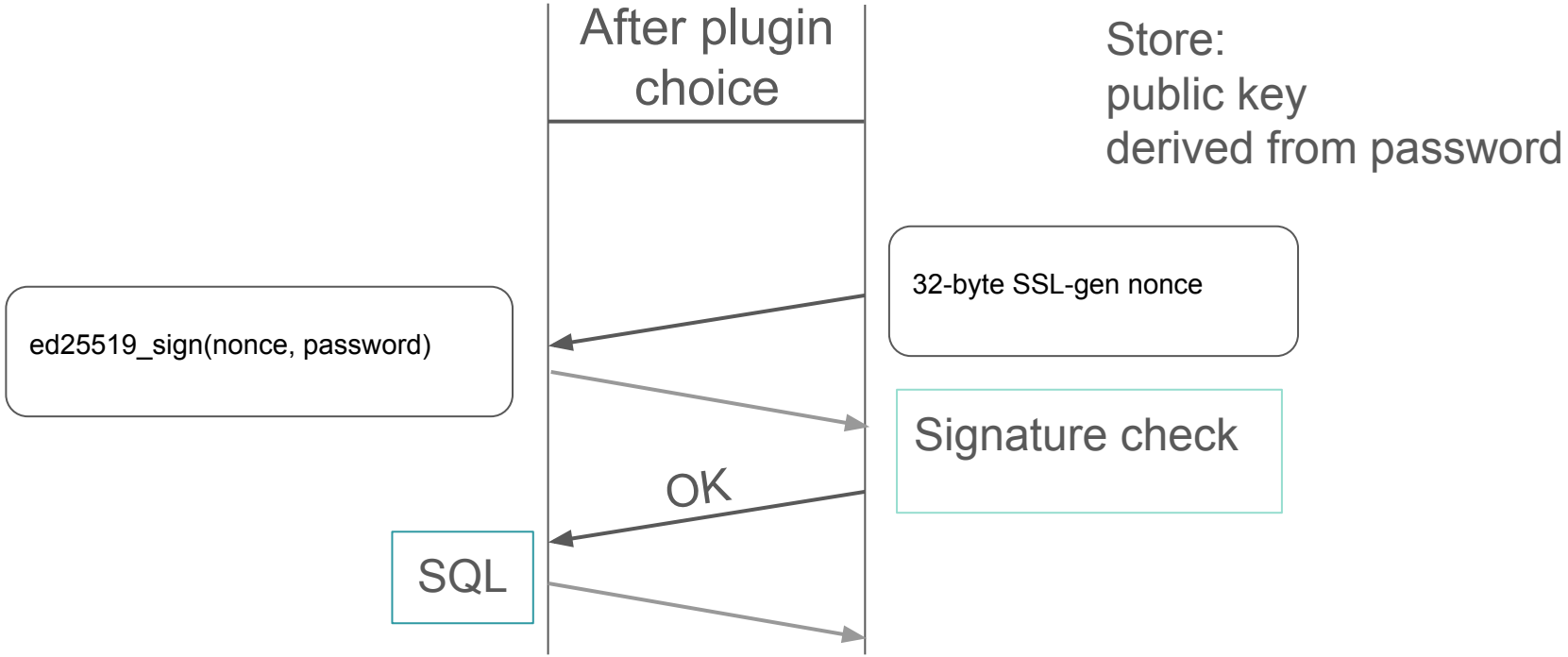
Case: native_pasword



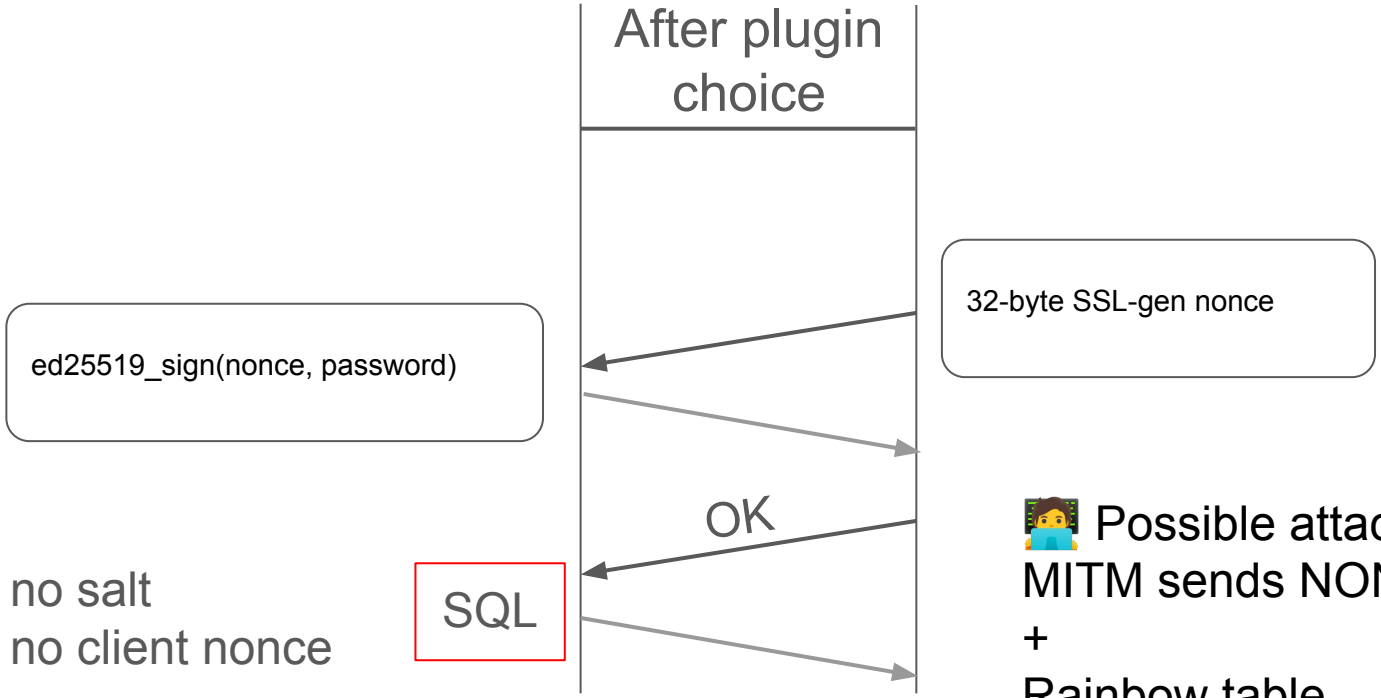
Case: ed25519



Case: ed25519



Case: ed25519

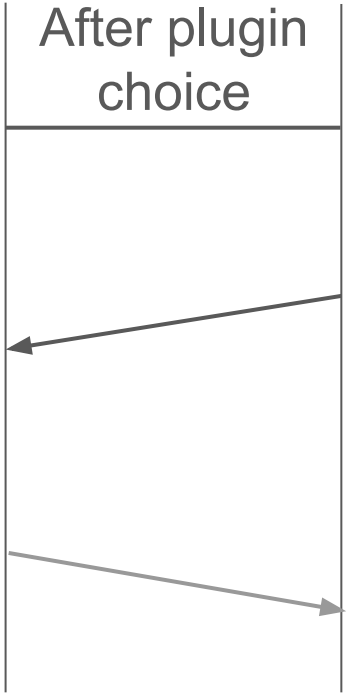


❌ no salt
❌ no client nonce

🤦 Possible attack:
MITM sends NONCE=0000
+
Rainbow table

New plugin: PARSEC

Store:
salt+Public key
derived from
PBKDF2(passord, salt,
iterations)



32-byte SSL-gen nonce
Derivation algorithm,
iterations
salt

Client Nonce
P = PBKDF2(password, salt, iter)
sign(nonce + client nonce, P)

✔️salt
✔️ nonce
✔️ standard ed25519 API

PARSEC:

✅ Uses OpenSSL, GnuTLS/nettle for ed25519, randoms, hashes

✅ Uses PBKDF2/SHA512 for salt with > 1k iterations

😓 windows: bcrypt/ncrypt doesn't support key derivation
=> We supply a portable implementation

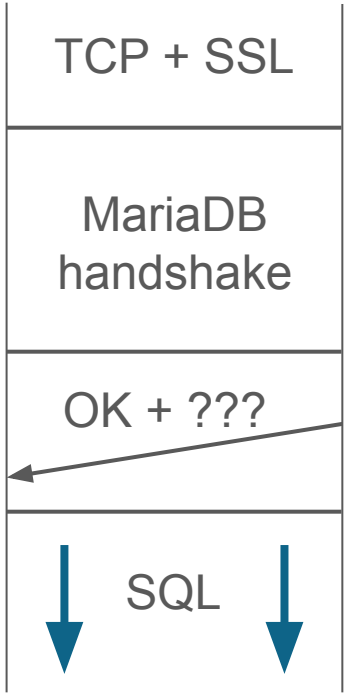
Conforms NIST recommendations:

- 32-byte scramble
- 32-byte nonce
- 18-byte salt
- 64-byte sig, key is 32-byte

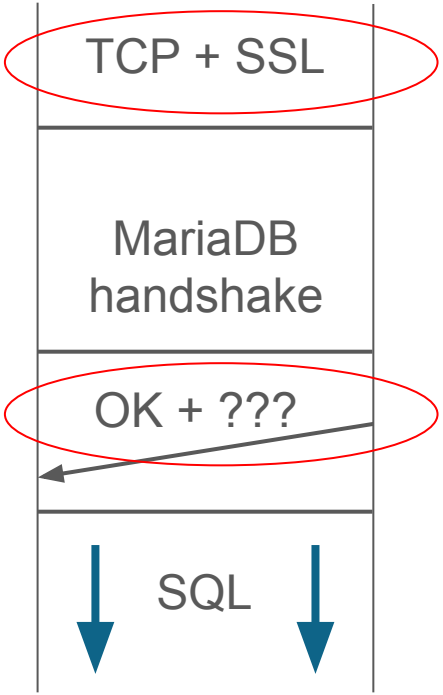
References

MariaDB

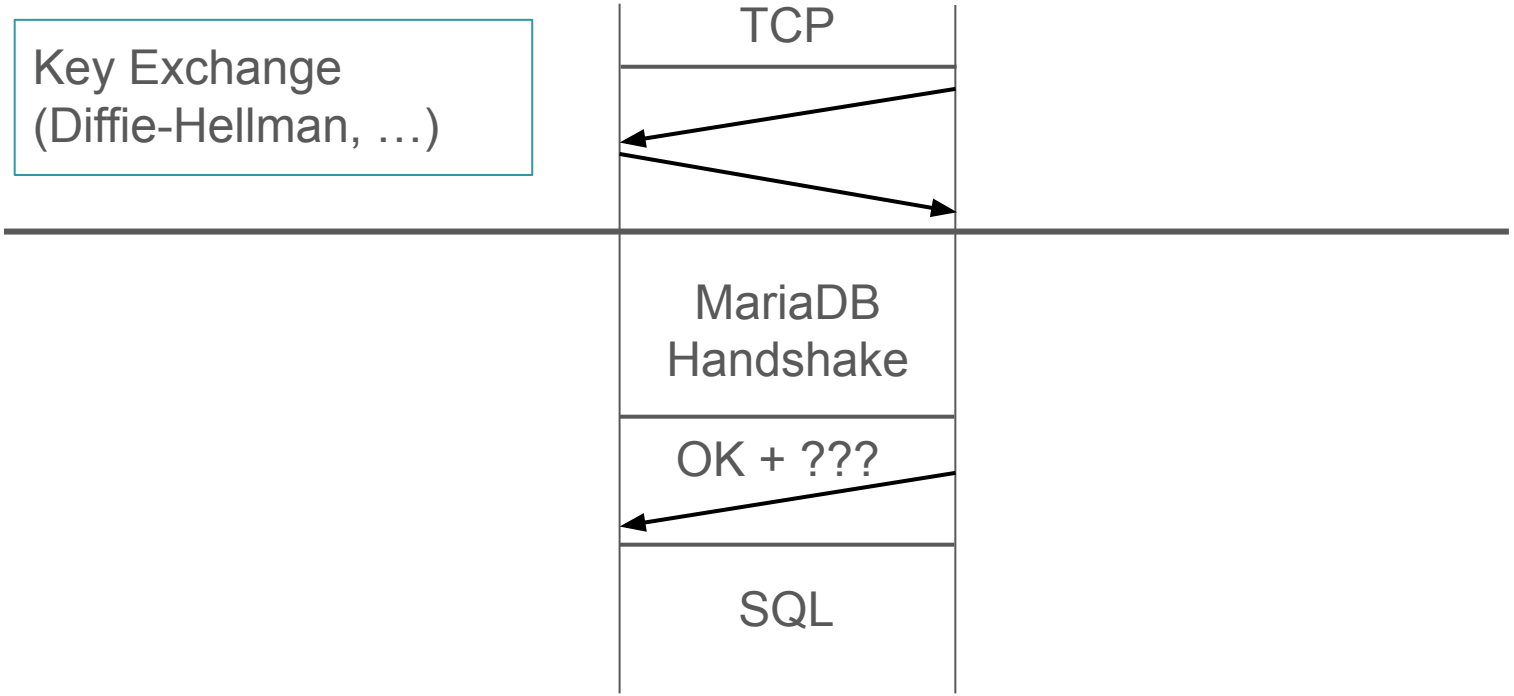
SSL/TLS: a closer look



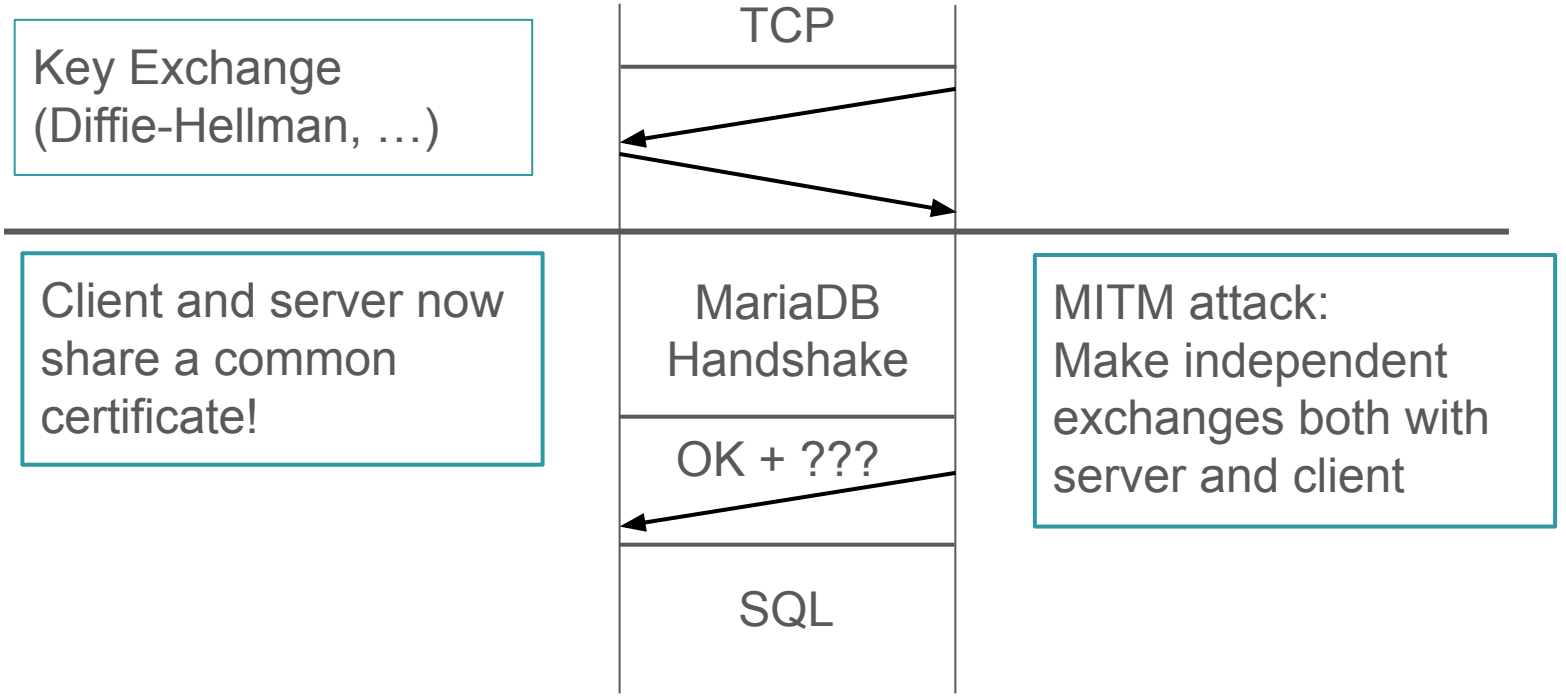
SSL/TLS: a closer look



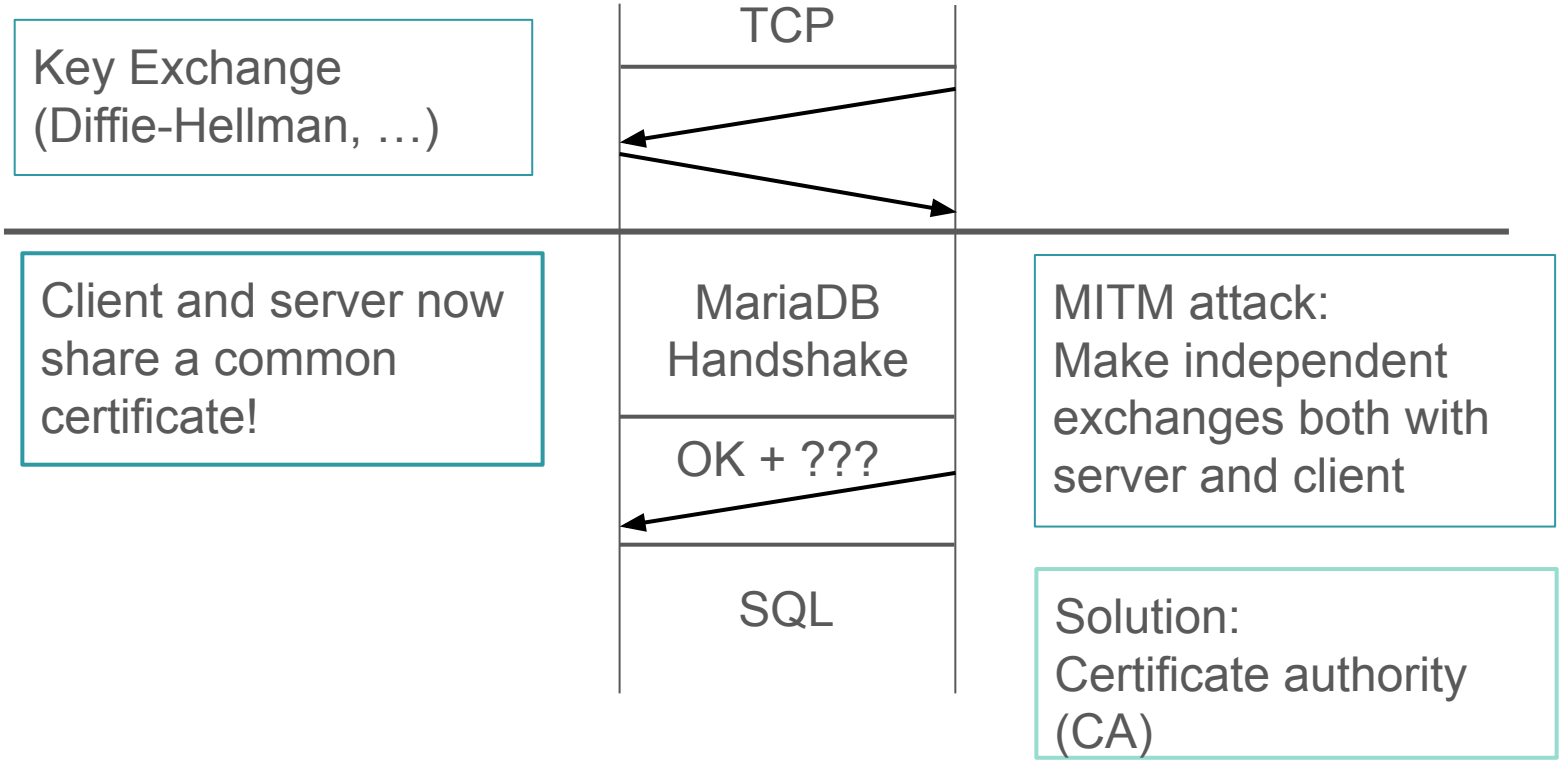
SSL/TLS: a closer look



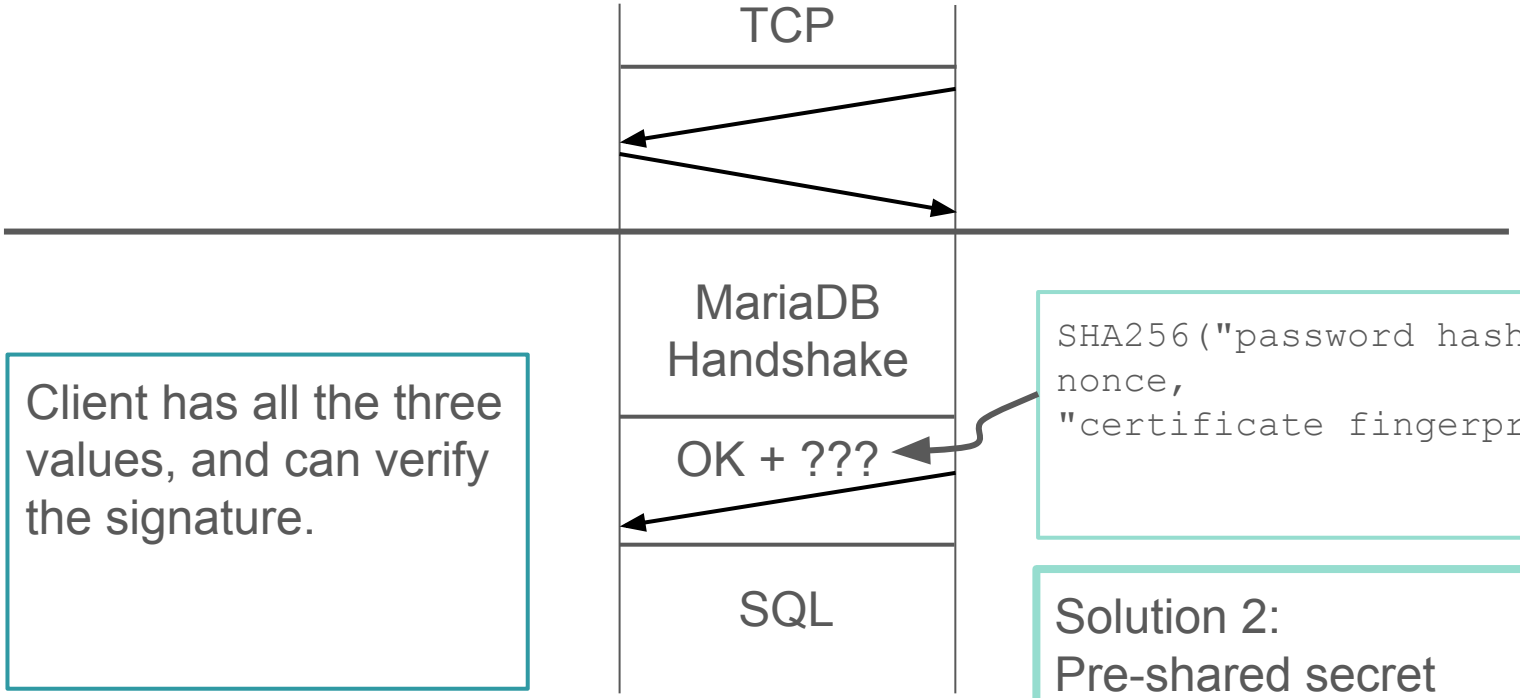
SSL/TLS: a closer look



SSL/TLS: a closer look



Zero-configuration SSL!



Client has all the three values, and can verify the signature.

```
SHA256("password hash",
nonce,
"certificate fingerprint")
```

Solution 2:
Pre-shared secret
(and we have it!)

Configuring Zero-config SSL

MariaDB Server:

- Nothing is required!

MariaDB clients:

Just enable SSL:

- Connector/J:

  `sslMode=verify-full`

- Connector/Node.js:

  `ssl: true`

- Connector/C

  Nothing is needed! It’s all set!

Nikita Maliavin

MariaDB Core Engineer
@MariaDB

Sergei Golubchik

Chief Architect
MariaDB Server
@MariaDB

MariaDB